REMARKS/ARGUMENTS

The present amendment is responsive to the final Office Action dated November 26, 2008. Claims 1, 4, 18, 19, 26, 29, 30, 32, 33, 36-43 and 45-50 have been amended. No new matter has been introduced by these amendments. Thus, claims 1, 3-4, 6-7, 9-26 and 28-50 are again presented for the Examiner's consideration in view of the following remarks. The rejections will be addressed in view of the claims as currently presented.

Reexamination and reconsideration of the above-identified application, pursuant to and consistent with 37 C.F.R. § 1.116 and in light of the following amendments and remarks, are respectfully requested. Good cause exists for the entry of this amendment in accordance with 37 C.F.R. § 1.116.

As an initial matter, applicants would like to thank the Examiner for the telephone interview with the undersigned on January 22, 2009. A number of issues were discussed, including aspects of the § 112 rejections. Turning to those rejections, claims 1, 3-4, 6-7, 9-26 and 28-50 were rejected under 35 U.S.C. § 112, second paragraph, as being indefinite. The rejection included 10 specific rejections (*i.e.*, rejections i - x) for different claims. As part of the telephone discussion with the Examiner, it is applicants' understanding that rejections iii, iv, v and vi would be withdrawn.

Rejections i and ii focused on "copying a task table" as recited in claims 1, 4 and 26. These claims have been modified in view of the discussion with the Examiner. Applicants submit that these claims are not indefinite and request that the rejections be withdrawn.

Rejection vii addressed claim 18, stating it was unclear why the task table is being copies from shared memory to local memory "if the task use is complete." (Office Action, p.3.) Claim 18 has been amended to clarify certain aspects of

the claim. Applicants submit that amended claim 18 is not indefinite and request that the rejections be withdrawn.

Rejections viii and ix pertain to claims 19 and 43. According to the Office Action, "it is unclear how the task table is being modified and what is defined by modifying the task table entry." (Office Action, p.3.) These claims have been amended as recited above. Applicants would also refer the Examiner to specification paragraph 0090, which discusses an example for a processor task yielding to another processor task for execution. In view of this, applicants submit that amended claims 19 and 43 are not indefinite and request that the rejections be withdrawn.

The final § 112 rejection pertains to claim 33. Here, the Office Action contends that "it is unclear where it is being copied to from the shared memory." (Office Action, p.3.) This claim has been amended to clarify that the processor tasks which are executed are copied from the shared memory to the processing units' local memories prior to execution. *See* also independent claim 29. In view of this, applicants submit that amended claim 33 is not indefinite and request that the rejection be withdrawn.

Before turning to the remaining grounds of rejection, applicants will address the Examiner's "Response to Arguments" as set forth on pages 16-17 of the final Office Action. This Response asserts that the term "operable" is "open ended thus indefinite." (*Id.* at p.17.) Applicants strongly disagree with this assertion.

It is respectfully submitted that the Court of Customs and Patent Appeals ("C.C.P.A.") held long ago "that there is nothing intrinsically wrong in defining something by what it does rather than by what it is." *In re Hallman*, 655 F.2d 212, 215 (C.C.P.A. 1981). Furthermore, "functional language in the claims ***must be given full weight and may not be disregarded*** in evaluating the patentability of the subject matter defined

employing such functional language" (*Ex parte Bylund*, 217 U.S.P.Q. 492 (Bd. App. 1981), emphasis added.)

A brief search of the USPTO's on-line patent database for the term "operable to" in the claims has revealed that no less than 168,364 issued patents include that limitation in the claims. While it is understood that some group art units presently discourage the use of this term, no evidence has been provided in the Office Action to support the contention that the term operable is indefinite. In fact, applicants submit the opposite is true. If there USPTO has a written policy or guideline stating why the term operable is indefinite, applicants request that such information be provided along with the next Office communication so that it may be evaluated on its merits.

Notwithstanding this, in an effort to expedite prosecution applicants have replaced the term "operable" with "configured" as discussed with the Examiner during the telephone interview. This is not done in view of a rejection and does not narrow the scope of the amended claims.

Turning to the rejections based upon prior art, claims 29-31 were rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,592,671 ("*Hirayama*"). Of these, claim 29 is independent. Applicants respectfully traverse the rejection.

Applicants respectfully disagree with the Office Action's assertions as to claim 29. Here, the Office Action contends that *Hirayama* teaches that the processing units "determine which of the processor tasks should be copied from the shared memory into their local memories and executed (col 2, lines 54-62; lines 64-67; col 5, lines 33-37; col 6, lines 9-16; lines 37-40)." (Office Action, p.8, numbered section 18.) Applicants respectfully disagree. What *Hirayama* actually states is:

> FIG. 1 is a block diagram of a resource management system of the first embodiment. Processors 100, 101, 102, . . . , 10n are connected to a common memory (or a shared memory) 12 via a bus 11, thereby forming a UMA (Uniform Memory Access)-type multiprocessor system, or a tightly coupled multiprocessor system. A process management table 13 contained in the common memory 12 is made up of process management table entries and is used to manage processes executed on the system. The individual process management table entries 130, 131, . . . , 13m are used to manage corresponding processes 140, 141, . . . , 14m, respectively. When there are executable processes on the system, those processes are linked to a run queue 15. Each processor selects a process to be executed from the processes linked to the run queue 15.

(*Hirayama*, col. 2, ll.54-62, 64-67.)

> The values of those priorities are expressed as high or low. For the initially set value for the priority, the value of the priority corresponding to a page on a local memory of a processor is made high, and the value of the priority corresponding to a page on a remote memory is made low.

(*Hirayama*, col. 5, ll.33-37.)

> Because the individual processors 500, 501, 502, . . . , 50n, when allocating pages to processes, select pages in the order of descending priorities, if processes move less frequently between processors, a processor will access a local memory more frequently, thereby enabling an efficient memory management. To cause processes to move less frequently between processors, a process management as explained in the first embodiment is effected.

(*Hirayama*, col. 6, ll.9-16.)

> If such a page is not present, the processor will call the page out demon and repeat step C1 (step C7). If such a page is present, the processor will allocate the page to a process. At step C5, if the value of the priority is high, the page being checked will also be allocated to a process (step C8).

(*Hirayama*, col. 6, ll.37-40.)

Applicants submit that none of the cited portions of *Hirayama* actually teaches what the Office Action asserts it teaches. The first cited portion describes the general block diagram of the resource management system, the process management table, the entries therein, and the run queue. The other cited portions describe how the system prioritizes and allocates the pages of the processor's local memory.

In contrast, claim 29 recites that the processing units are configured to use the task table to determine which of the processor tasks should be copied from the shared memory into the local memories and be executed. This is not what *Hirayama* teaches.

The final rejection reiterated the position set forth in the first Office Action. The final rejection also states:

> The claimed limitations if brad [*sic*] and does not specify what is the criteria to determine which task should be copied and moreover in line 4 and 8 claim recites "operable" which is open ended thus indefinite and does mandate the use of shared memory or task table to determine or copy task tasks [*sic*] from shared to local memory.
>
> (Final Office Action, p.17, numbered section 46.)

Applicants have addressed the issue with regard to the term "operable" above. Furthermore, claim 29 has been amended to recite, in part, "wherein the processing units are configured to use the task table to determine which of the processor tasks should be copied from the shared memory into their local memories and executed and are configured to maintain and modify the task table during execution of the processor tasks." Applicants submit that *Hirayama* does not teach or suggest all such limitations as claimed.

In view of the above, applicants submit that independent claim 29 is patentable over *Hirayama*. Furthermore, claims 30-31 depend from claim 29 and contain all the

limitations thereof. For at least this reason, the subject dependent claims are likewise patentable over *Hirayama*.

Claims 1, 3-4, 6-7, 9-14, 16-22, 26, 28, 32-39 and 41-47 were rejected under 35 U.S.C. § 103(a) as being unpatentable over *Hirayama* in view of EP 0459931 ("*Bahr*"). Of these, claims 1, 4, 19, 26 and 43 are independent. Applicants respectfully traverse the rejection.

The final Office Action asserts that *Bahr* "teaches copying a task table from the shared memory to the local memory of a processing unit" and "copying the task table from the local memory of a processing unit to the shared memory." (Final Office Action, p.6, numbered section 11.) Applicants respectfully disagree. The rejection cites to several sections of *Bahr* which are reproduced below.

> (b) for each data operation having a specific affinity for a selected one of the processing devices, either (i) assigning the data operation to the selected processing device if the processing device is available for the assignment
>
> (Col.3, ll.20-24.)

> When an IMPI task is switched into processor 18 for execution, data and IMPI instructions pertinent to the task are loaded into registers 34 either from main storage memory 28 or from the processor's cache memory 36. When the task is switched out of processor 18, processor data is stored to the main storage memory, and also may be stored to the cache memory, if the associated cache lines are valid. Processing device 18 further includes a one-way associative store-through cache memory 36, which includes a plurality of cache lines, each associated with a particular set of addresses or locations in the main storage memory. Cache memory 36 further includes a cache directory 38. The cache directory is a record of the main storage memory addresses mapped into the cache memory, and stores an indication of whether each corresponding location in the cache memory is valid or invalid.
>
> (Col.6, ll.27-43.)

> Figs. 4-6 are sections of a flow chart useful in understanding the manner in which the task dispatching program and task dispatching queue are used to allocate

> tasks to the processing devices. In general, the TDEs [Task Dispatching Elements] are scanned in order of descending priority, and either assigned to one of the processors, bypassed, or reserved for subsequent assignment.
>
> (Col.9, ll.6-12.)

While *Bahr* does disclose scanning task dispatching elements in order of descending priority and may be assigned to a processor, this is not what is claimed. Applicants submit that the TDEs, as best understood, are not task tables, task table entries or task queues as claimed. Furthermore, the cited portions of *Bahr* do not disclose that the TDEs are copied from shared memory to local memory or from local memory to shared memory. Thus, applicants submit that there are insufficient facts to support the Examiner's contentions regarding *Bahr*. In view of this, applicants also submit that *Bahr* does not overcome the admitted deficiencies of *Hirayama* and a *prima facie* case for obviousness has not been met.

Therefore, independent claims 1, 4, 19, 26 and 43 are in condition for allowance. Furthermore, claims 3, 6-7, 9-14, 16-18, 20-22, 28, 32-39, 41-42 and 44-47 depend from these independent claims, respectively, and contain all the limitations thereof. Thus, for at least this reason, applicants submit the subject dependent claims are likewise in condition for allowance.

In addition, the rejected dependent claims are separately patentable over the applied combination. For instance, the rejection of claim 16 asserts that *Bahr* "teaches each of the task table entries includes an indication as to whether the associated processor task is READY to be executed or RUNNING on one or more of the sub-processing units (col 1, lines 34-43); and the method further includes modifying the given task table entry to indicate that the given processor task is RUNNING (col 1, lines 43-47)." (Final Office Action at pp.10-11, numbered section 23. What *Bahr* actually states is:

> Each of the processors may be assigned to the highest priority task available when not executing a task of higher priority. A processor identification is employed to prevent other processors from gaining access to the same task. In U.S. Patent No. 4,807,281, a processor can store a protective code corresponding to its address at a location associated with a particular job or task, but not if another processor already has entered its code at the job. A processor, upon detecting the protective code of another processor associated with the task data, forwards its own task to the other processor to reroute the job, to prevent a single job from running on more than one processor simultaneously.
>
> (Col.1, ll.34-47.)

Nothing in the cited portion of *Bahr* supports the rejection of dependent claim 16. By way of example, there is no discussion of whether a processor task is in a READY state or a RUNNING state. And while *Bahr* mentions storing a protective code corresponding to an address at a location associated with a particular job or task, there is no discussion of task table entries or modification thereof as claimed. Thus, claim 16 is patentable over *Hirayama* and *Bahr*.

With regard to claim 18, the rejection asserts that *Bahr* "teaches permitting the task queue and the task table to be copied from the shared memory to the local memory of the given sub-processing unit when the given sub-processing unit has completed its use thereof (col 10, lines 41-49)." (Final Office Action, p.11, numbered section 25.

Claim 18 has been amended to recite "locking the task table prior to copying from the shared memory to the local memory of the given sub-processing unit, and unlocking the task table when the given sub-processing unit has completed its use thereof." Applicants submit that *Bahr* fails to teach or otherwise suggest such limitations. Furthermore, the cited portion of *Bahr* actually states:

> At the beginning of the reverse scan, the Assigned Task Count is compared with a "usable" processing device count

> (Usable Count) equal to the lesser of (i) the number of processing devices and (ii) the number of tasks in the task dispatching queue. If, in view of the comparison at 140, the Assigned Task Count equals the Usable Count, the allocation of tasks is complete, and the tasks are dispatched, each to its associated one of processing devices 18-24.
>
> (Col.10, ll.41-49.)

Nothing in this section of *Bahr* supports the assertions in the rejection of claim 18. There is no copying of a task table, or locking or unlocking of same. Thus, claim 18 is patentable over *Hirayama* and *Bahr*.

And claim 20 was rejected because *Bahr* purportedly discloses a "task queue including at least one of a head pointer and a tail pointer, the head pointer providing an indication of a new first one of the processor tasks in the list of processor tasks, and the tail pointer providing an indication of a last one of the processor tasks in the list of processor tasks ... and permitting the processing units to use the task table and the task queue to determine which of the processor tasks should be executed next." (Final Office Action, pp.12-13, numbered section 27.) Applicants respectfully disagree. While *Bahr* discusses pointers referencing task dispatching elements of different priorities, this is not what is claimed. Therefore, applicants submit that a *prima facie* case for obviousness has not been met, and claim 20 is patentable over *Hirayama* and *Bahr*.

Claims 15, 23-25, 40 and 48-50 were rejected under 35 U.S.C. § 103(a) as being unpatentable over *Hirayama* in view of *Bahr* and U.S. Patent No. 6,321,308 ("*Arnon*"). All of these claims are dependent.

Applicants respectfully traverse the rejection. *Arnon* does not overcome the deficiencies of *Hirayama* and *Bahr*. And as these claims depend from the aforementioned independent claims and contain all the limitations thereof applicants submit the

subject dependent claims are likewise in condition for allowance for at least the reasons set forth above.

In view of the above, each of the presently pending claims in this application is believed to be in immediate condition for allowance. Accordingly, the Examiner is respectfully requested to withdraw the outstanding rejection of the claims and to pass this application to issue.

If, however, for any reason the Examiner does not believe that such action can be taken at this time, it is respectfully requested that he/she telephone applicant's attorney at (908) 654-5000 in order to overcome any additional objections which he might have.

If there are any additional charges in connection with this requested amendment, the Examiner is authorized to charge Deposit Account No. 12-1095 therefor.

Dated: May 12, 2009

Respectfully submitted,

Electronic signature: /Andrew T. Zidel/

Andrew T. Zidel
Registration No.: 45,256
LERNER, DAVID, LITTENBERG,
KRUMHOLZ & MENTLIK, LLP
600 South Avenue West
Westfield, New Jersey 07090
(908) 654-5000
Attorney for Applicant

1013225_1.DOC